🛠️ Ejercicio Práctico – Nivel Medio

📌 **Título:** *Realización de Pruebas de Penetración en OWASP Juice Shop*

🎯 Objetivo del ejercicio:

Realizar **pruebas de penetración** avanzadas en una aplicación web vulnerable (OWASP Juice Shop) para identificar vulnerabilidades de seguridad en diferentes capas de la aplicación. Utilizarás **OWASP ZAP** o **Burp Suite** para automatizar y realizar pruebas manuales, además de aprender a mitigar estas vulnerabilidades.

📝 Escenario:

El entorno vulnerable de **OWASP Juice Shop** está diseñado para enseñar sobre las mejores prácticas de seguridad en aplicaciones web. En este ejercicio, deberás realizar pruebas de penetración avanzadas con el objetivo de identificar y explotar vulnerabilidades en la aplicación.

🔍 Tu tarea:

Paso 1 – Acceso a la Aplicación:

1. **Inicia OWASP Juice Shop** en tu máquina local o en un entorno controlado.
   Si aún no tienes la aplicación instalada, puedes acceder a su página oficial y seguir las instrucciones de instalación.

Paso 2 – Exploración de la Aplicación:

1. Navega a través de las diversas secciones de la aplicación y realiza las siguientes acciones:
    - Accede a las páginas de registro y de inicio de sesión, intentando realizar **inyecciones SQL** en los campos de entrada.

- Prueba **Cross-Site Scripting (XSS)** en los formularios y campos de búsqueda para verificar si la aplicación es vulnerable a este tipo de ataque.

- Revisa las solicitudes de red para **datos sensibles** que podrían ser expuestos, como contraseñas o tokens de autenticación.

Paso 3 – Realización de Pruebas con Herramientas de Seguridad:

1. **Escaneo automático con OWASP ZAP**:

   - Realiza un escaneo automatizado en la aplicación usando **OWASP ZAP**. Deberías detectar vulnerabilidades comunes como **XSS**, **inyección SQL**, **Cross-Site Request Forgery (CSRF)** y más.

2. **Interceptación y Modificación de Solicitudes con Burp Suite**:

   - Configura **Burp Suite** para interceptar y modificar solicitudes HTTP entre el navegador y el servidor. Esto te permitirá probar manipulaciones de parámetros y encontrar vulnerabilidades como **inyección de comandos** o **cambio de roles**.

Paso 4 – Explotación de Vulnerabilidades:

1. **Explotar XSS**:
   Intenta realizar un ataque XSS para robar información de sesión (cookies) o realizar otras acciones maliciosas en la aplicación.

2. **Explotar Inyección SQL**:
   Si has encontrado un campo vulnerable a **inyección SQL**, intenta realizar un ataque para acceder a la base de datos. Usa técnicas como `' OR 1=1 --` para obtener acceso no autorizado.

3. **Explotar CSRF**:
   Intenta realizar un ataque **Cross-Site Request Forgery (CSRF)** creando un enlace malicioso que permita a un usuario autenticado realizar una acción no autorizada en su cuenta, como cambiar su dirección de correo electrónico.

Paso 5 – Mitigación de Vulnerabilidades:

1. Para cada vulnerabilidad identificada, documenta cómo podría ser **mitigada**. Algunas soluciones comunes incluyen:

   - **Escapar las entradas del usuario** para prevenir **XSS**.

   - **Uso de declaraciones preparadas** para prevenir **inyección SQL**.

    - **Validación de entradas** y **tokens CSRF** para proteger contra ataques de CSRF.

2. **Propuesta de medidas de seguridad adicionales**, como el uso de **WAF (Web Application Firewalls)** y **autenticación multifactor**.

Paso 6 – Informe Final:

1. **Documenta los resultados** de las pruebas de penetración realizadas:
    - **Vulnerabilidades encontradas**.
    - **Explotación de vulnerabilidades** (si fue posible).
    - **Soluciones recomendadas** para cada vulnerabilidad.
2. **Proporciona un informe detallado** con capturas de pantalla de las herramientas utilizadas y los ataques realizados.

✅ Resultado esperado:

- **Vulnerabilidades identificadas**: Como mínimo, debes encontrar y explotar 3 vulnerabilidades diferentes en la aplicación.
- **Análisis de impacto**: Explica cómo cada vulnerabilidad podría ser explotada y el impacto potencial en la aplicación.
- **Soluciones recomendadas**: Describe cómo mitigar las vulnerabilidades encontradas.
- **Informe detallado**: Un documento con las pruebas realizadas, los ataques ejecutados, y las soluciones propuestas.

📄 Entrega sugerida:

- Capturas de pantalla de las vulnerabilidades encontradas y cómo fueron explotadas.
- Documento con las soluciones recomendadas y las medidas de seguridad a implementar.

- Resultados de las pruebas con herramientas como **OWASP ZAP** o **Burp Suite**.
- Informe final con detalles sobre los ataques realizados y sus impactos potenciales.

🧰 Herramientas recomendadas:

- **OWASP Juice Shop** (Aplicación vulnerable)
- **OWASP ZAP** (Herramienta para escaneo de vulnerabilidades)
- **Burp Suite** (Herramienta para interceptar y modificar solicitudes HTTP)
- **Google Chrome Developer Tools** (Para inspección y modificación de tráfico)
